意味分割のための完全畳み込みネットワーク

Jonathan Long* Evan Shelhamer* Trevor Darrell
UC Berkeley
{jonlong,shelhamer,trevor}@cs.berkeley.edu

Abstract

畳み込みネットワークは、特徴の階層をもたらす強力な視覚モデルである。エンドツーエンドで学習されたピクセルからピクセルへの畳み込みネットワークが、セマンティックセグメンテーションの最先端技術を上回ることを示す。我々の重要な洞察は、任意のサイズの入力を受け取り、それに対応するサイズの出力を効率的な推論と学習で生成する「完全畳み込み」ネットワークを構築することである。完全畳み込みネットワークの空間を定義し、詳細に説明し、空間的に密な予測タスクへの適用を説明し、先行モデルとの関連性を導き出す。現代の分類ネットワーク(AlexNet [19]、VGGネット [31]、GoogLeNet [32])を完全畳み込みネットワークに適応させ、微調整[4]によって学習した表現をセグメンテーションタスクに転送する。次に、深く粗い層からの意味情報と、浅く細かい層からの外観情報を組み合わせて、正確で詳細なセグメンテーションを生成する、新しいアーキテクチャを定義する。我々の完全畳み込みネットワークは、PASCAL VOC(2012年の平均IU62.2%に対して20%の相対的改善)、NYUDv2、SIFT Flowの最先端のセグメンテーションを達成し、推論は典型的な画像に対して5分の1秒未満しかかからない。

1. Introduction

畳み込みネットワークは認識の進歩を牽引している。Convnetsは全画像分類[19, 31, 32]を改善するだけでなく、構造化された出力を持つ局所的なタスクも進歩している。これらには、バウンディングボックスオブジェクト検出[29, 12, 17]、パーツとキーポイント予測[39, 24]、局所対応[24, 9]の進歩が含まれる。

粗い推論から細かい推論への進行における自然な次のステップは、すべてのピクセルで予測を行うことである。先行するアプローチでは、セマンティックセグメンテーションにコンブネットを使用しており[27, 2, 8, 28, 16, 14, 11]、各ピクセルはその包含するオブジェクトまたは領域のクラスでラベル付けされるが、この研究が対処する欠点がある。

*Authors contributed equally

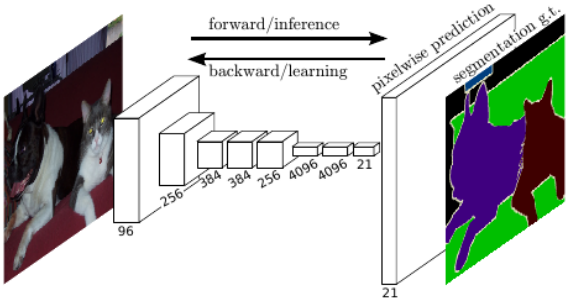

図1. 完全畳み込みネットワークは、セマンティックセグメンテーションのようなピクセル単位のタスクに対して、高密度の予測を効率的に学習することができる。

我々は、セマンティックセグメンテーションでエンドツーエンド、ピクセルツーピクセルで学習された完全畳み込みネットワーク(FCN)が、さらなる機械なしで最先端を超えることを示す。我々の知る限り、これは(1)ピクセル単位の予測、(2)教師ありの事前学習から、FCNをエンドツーエンドで学習する最初の研究である。既存のネットワークの完全畳み込みバージョンは、任意のサイズの入力から密な出力を予測する。学習と推論は、密なフィードフォワード計算とバックプロパゲーションにより、全画像-データ時間実行される。ネットワーク内アップサンプリング層は、サブサンプリングプーリングによるネットのピクセル単位の予測と学習を可能にする。

この方法は漸近的にも絶対的にも効率的であり、他の研究における複雑さの必要性を排除している。パッチワイズ学習は一般的であるが[27, 2, 8, 28, 11]、完全畳み込み学習の効率性に欠ける。我々のアプローチは、スーパーピクセル[8, 16]、提案[16, 14]、ランダムフィールドや局所分類器によるポストホックリファインメント[8, 16]などの前処理と後処理の複雑さを利用しない。我々のモデルは、分類ネットを完全に畳み込み、学習された表現から微調整するものとして再解釈することで、分類における最近の成功[19, 31, 32]を密な予測に移行する。一方、先行研究では、教師ありの事前学習なしで小さな畳み込みネットを適用している[8, 28, 27]。

意味的セグメンテーションは、意味論と位置の間の固有の緊張に直面する:グローバルな情報は何を解決し、ローカルな情報はどこで解決するか。

深い特徴階層は、ローカルからグローバルへのピラミッドにおいて、位置とセマンティクスを共同で符号化する。我々は、セクション4.2において、深い粗い意味情報と浅い細かい外観情報を組み合わせる新しい「スキップ」アーキテクチャを定義する(図3参照)。

次のセクションでは、深層分類ネット、FCN、および畳み込みネットを用いたセマンティックセグメンテーションへの最近のアプローチに関する関連研究をレビューする。以下のセクションでは、FCNの設計と密な予測トレードオフを説明し、ネットワーク内アップサンプリングと多層の組み合わせによる我々のアーキテクチャを紹介し、我々の実験フレームワークを説明する。最後に、PASCAL VOC 2011-2、NYUDv2、SIFT Flowで最先端の結果を示す。

2. Related work

我々のアプローチは、画像分類[19, 31, 32]と転移学習[4, 38]のためのディープネットの最近の成功を利用している。転送は、まず様々な視覚認識タスク[4, 38]で実証され、次に検出、そしてハイブリッド提案分類器モデル[12, 16, 14]のインスタンスとセマンティックセグメンテーションの両方で実証された。次に、セマンティックセグメンテーションを直接、高密度に予測するために、分類ネットを再アーキテクチャーし、微調整する。FCNの空間を図示し、このフレームワークにおける過去のモデルと最近のモデルを位置づける。

完全畳み込みネットワーク Matanら[25]は、古典的なLeNet[21]を拡張して数字の文字列を認識した。彼らのネットは1次元の入力文字列に限定されていたため、Matanらはビタビ復号を使って出力を得た。WolfとPlatt[37]は、convnetの出力を、郵便番号ブロックの4つのコーナーの検出スコアの2次元マップに拡張した。これらの歴史的研究はいずれも、検出のために推論と完全畳み込み学習を行っている。Ningら[27]は、線虫組織の粗い多クラスセグメンテーションのための畳み込みネットを、完全畳み込み推論で定義している。

完全畳み込み計算は、多層ネットの現代においても利用されている。Sermanetら[29]によるスライディングウィンドウ検出、PinheiroとCollobert[28]によるセマンティックセグメンテーション、Eigenら[5]による画像復元は、完全畳み込み推論を行う。完全畳み込み学習はまれであるが、Tompsonら[35]は、この方法の説明や分析はしていないが、ポーズ推定のためのエンドツーエンドのパーツ検出器と空間モデルを学習するために効果的に使用している。

あるいは、Heら[17]は、分類ネットの非畳み込み部分を破棄して特徴抽出器を作る。提案と空間ピラミッドプーリングを組み合わせることで、分類のための局所的で固定長の特徴を得る。高速で効果的ではあるが、このハイブリッドモデルはエンドツーエンドで学習することはできない。

convnetsを用いた高密度予測 Ningら[27]、Farabetら[27]によるセマンティックセグメンテーションを含む、高密度予測問題にconvnetsを適用した最近の研究がいくつかある。

[8]、Pinheiro and Collobert [28]、Ciresan らによる電子顕微鏡の境界予測 [2]、Ganin and Lempitsky [11]によるニューラルネット/最近傍ハイブリッドモデルによる自然画像の境界予測、Eigen らによる画像復元と深度推定 [5, 6].これらのアプローチに共通する要素は以下の通りである。

- 容量と受容野を制限する小さなモデル;パッチワイズ
- トレーニング [27, 2, 8, 28, 11];
- スーパーピクセル投影、ランダムフィールド正則化、フィルタリング、局所分類による後処理 [8, 2, 11];
- OverFeat[29]によって導入された、密な出力のための入力シフトと出力インターレース[28, 11];
- マルチスケールピラミッド処理 [8, 28, 11];
- 飽和タン非線形性 [8, 5, 28]; および
- ensembles [2, 11],

一方、我々の手法はこの機械を使わずに行う。しかし、我々はパッチワイズ学習3.4と「shift-and-stitch」密出力3.2をFCNの観点から研究している。また、Eigenら[6]による完全連結予測が特殊なケースである、ネットワーク内アップサンプリング3.3についても議論する。

これらの既存の手法とは異なり、我々は、教師ありの事前学習として画像分類を使用し、深層分類アーキテクチャを適応・拡張し、画像全体の入力と画像全体の接地スルーからシンプルかつ効率的に学習するために、完全に畳み込み的に微調整を行う。

Hariharanら[16]とGuptaら[14]も同様に、ディープ分類ネットをセマンティックセグメンテーションに適応させるが、提案と分類のハイブリッドモデルでは適応させる。これらのアプローチは、検出、セマンティックセグメンテーション、インスタンスセグメンテーションのためのバウンディングボックスと/または領域提案をサンプリングすることで、R-CNNシステム[12]を微調整する。どちらの方法もエンドツーエンドで学習されない。

彼らはそれぞれPASCAL VOCセグメンテーションとNYUDv2セグメンテーションで最先端の結果を達成しているので、セクション5で我々のスタンドアロン、エンドツーエンドFCNと彼らのセマンティックセグメンテーションの結果を直接比較する。

3. 完全畳み込みネットワーク

convnetの各データ層は、h×w×dの大きさの3次元配列であり、hとwは空間次元、dは特徴またはチャネル次元である。最初の層は画像で、ピクセルサイズはh×w、カラーチャンネルはdである。上位層の位置は、それらがパス接続された画像内の位置に対応し、それらはそれらの受容野と呼ばれる。

Convnetsは翻訳不変性に基づいて構築されている。その基本構成要素(畳み込み、プーリング、活性化関数)は局所的な入力領域で動作し、相対的な空間座標にのみ依存する。特定層の位置(i, j)のデータベクトルをx_{ij}、次の層をy_{ij}と書くと、

これらの関数は出力y_{ij}を次のように計算する。

$$\mathbf{y}_{ij} = f_{ks}\left(\{\mathbf{x}_{si+\delta i, sj+\delta j}\}_{0 \le \delta i, \delta j \le k}\right)$$

ここで、kはカーネルサイズ、sはストライドまたはサブサンプリング係数、f_{ks}は層のタイプを決定する:畳み込みまたは平均プーリングのための行列乗算、最大プーリングのための空間最大、または活性化関数のための要素ごとの非線形性、そして他のタイプの層について、など。
この関数形は、カーネルサイズとストライドが変換ルールに従う合成のもとで維持される。

$$f_{ks} \circ g_{k's'} = (f \circ g)_{k'+(k-1)s', ss'}.$$

一般的なディープネットが一般的な非線形関数を計算するのに対して、この形式の層のみを持つネットは非線形フィルタを計算する。FCNは、任意のサイズの入力に対して自然に動作し、対応する(おそらく再サンプリングされた)空間次元の出力を生成する。
FCNで構成される実数値損失関数はタスクを定義する。損失関数が最終層の空間次元の和、$\ell(x;\theta) = \sum_{ij} \ell'(x_{ij};\theta)$であれば、その勾配は各空間成分の勾配の和となる。したがって、画像全体に対して計算されたℓの確率的勾配降下は、最終層の受容野をすべてミニバッチとして、ℓ'の確率的勾配降下と同じになる。

これらの受容野が大きく重なる場合、フィードフォワード計算とバックプロパゲーションの両方が、独立にパッチごとに計算するのではなく、画像全体にわたってレイヤーごとに計算する方がはるかに効率的である。
次に、分類ネットを粗い出力マップを生成する完全畳み込みネットに変換する方法を説明する。ピクセル単位の予測では、これらの粗い出力をピクセルに結びつける必要がある。3.2節では、OverFeat [29]がこの目的のために導入したトリックについて説明する。このトリックを等価なネットワーク修正として再解釈することで、洞察を得ることができる。効率的で効果的な代替案として、セクション3.3でアップサンプリングのためのデコンボリューション層を紹介する。セクション3.4では、パッチワイズサンプリングによる学習を検討し、セクション4.3で、我々の画像全体の学習がより高速で、同様に効果的であることを示す。

3.1. 密な予測のための分類器の適応

LeNet[21]、AlexNet[19]、そしてその深い後継者[31, 32]を含む典型的な認識ネットは、表向きは固定サイズの入力を受け取り、非空間出力を生成する。これらのネットの完全連結層は、固定された次元を持ち、空間座標を捨てる。しかし、これらの完全連結層は、入力領域全体をカバーするカーネルを持つ畳み込みと見なすこともできる。そうすることで、どのようなサイズの入力も取り、分類マップを出力する完全畳み込みネットワークに落とし込む。

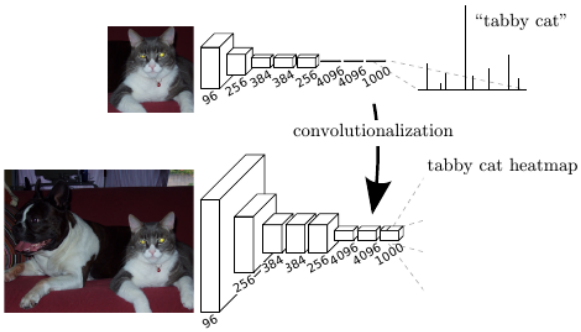

図2. 完全連結層を畳み込み層に変換することで、分類ネットはヒートマップを出力することができる。図1のように)レイヤーと空間損失を追加することで、エンドツーエンドの高密度学習のための効率的なマシンが生成される。

この変換は図2に示されている(対照的に、Leら[20]のような非畳み込みネットは、この能力を欠いている)。
さらに、得られたマップは特定の入力パッチに対する元のネットの評価と同等であるが、計算はそれらのパッチの重複領域に対して高度に償却される。例えば、AlexNetが227×227の画像の分類スコアを生成するのに1.2ms(典型的なGPU上)かかるのに対し、完全畳み込みバージョンは500×500の画像から10×10グリッドの出力を生成するのに22msかかり、これはナイーブアプローチ[1]の5倍以上速い。
これらの畳み込みモデルの空間出力マップは、セマンティックセグメンテーションのような高密度な問題には自然な選択となる。各出力セルでグランドトゥルースが利用可能であるため、フォワードパスとバックワードパスの両方が簡単であり、どちらも畳み込みの固有の計算効率(および積極的な最適化)を利用する。
AlexNetの例では、単一画像で2.4ms、完全畳み込み10×10出力マップで37msとなり、フォワードパスと同様の高速化が実現された。この密なバックプロパゲーションを図1に示す。
分類ネットを完全畳み込みとして再解釈すると、どのようなサイズの入力に対しても出力マップが得られるが、出力次元は通常サブサンプリングによって減少する。分類ネットは、フィルタを小さくし、計算量を合理的に保つためにサブサンプリングする。これは、これらのネットの完全畳み込みバージョンの出力を粗くし、入力のサイズから出力ユニットの受容野のピクセルストライドに等しいファクターだけ減少させる。

[1]Assuming efficient batching of single image inputs. The classification scores for a single image by itself take 5.4 ms to produce, which is nearly 25 times slower than the fully convolutional version.

3.2. シフトアンドスティッチはフィルタの希薄化

入力シフトと出力インターレースは、OverFeat [29]によって導入された、補間なしで粗い出力から密な予測を得るトリックである。出力がf倍ダウンサンプリングされた場合、入力は(左と上のパディングによって)xピクセル右にシフトされ、yピクセル下にシフトされる。, f − 1} × {0,.., f − 1}.これらのf^2個の入力はそれぞれconvnetにかけられ、予測値がその受容野の中心にある画素に対応するように出力はインターレースされる。

convnetのフィルタとレイヤーストライドだけを変更することで、このshift-and-stitchトリックと同じ出力を生成できる。入力ストライドsの層(畳み込みまたはプーリング)と、フィルタ重みf_{ij}(特徴次元を消去する、ここでは無関係)を持つ次の畳み込み層を考える。下層の入力ストライドを1に設定すると、シフト&スティッチと同様に、その出力がs倍にアップサンプリングされる。しかし、元のフィルタをアップサンプリングされた出力で畳み込んでも、元のフィルタはその(現在はアップサンプリングされた)入力の縮小部分しか見ていないため、トリックと同じ結果は得られない。このトリックを再現するために、フィルターを次のように拡大してレアフィリングする。

$$f'_{ij} = \begin{cases} f_{i/s,j/s} & \text{sがiとjの両方を分割する場合、} \\ 0 \end{cases}$$

(iとjはゼロベース)。トリックの完全なネット出力を再現するには、すべてのサブサンプリングが除去されるまで、このフィルター拡大をレイヤーごとに繰り返す必要がある。

ネット内のサブサンプリングを減らすことはトレードオフである:フィルターはより細かい情報を見るが、受容野が小さく、計算に時間がかかる。シフト・アンド・スティッチのトリックはもう一つのトレードオフであることを見てきた。フィルタの受容野サイズを小さくすることなく出力を密にするが、フィルタは元の設計よりも細かいスケールで情報にアクセスすることは禁止されている。

我々はshift-and-stitchの予備実験を行ったが、我々のモデルでは使用しない。次節で述べるように、アップサンプリングによる学習は、特に後述するスキップレイヤーフュージョンと組み合わせることで、より効果的かつ効率的であることが分かる。

3.3. アップサンプリングは逆方向ストライド畳み込み

粗い出力を密なピクセルに接続するもう一つの方法は補間である。例えば、単純なバイリニア補間は、入力セルと出力セルの相対位置にのみ依存する線形マップによって、最も近い4つの入力から各出力y_{ij}を計算する。

ある意味で、係数fによるアップサンプリングは、1/fの分数入力ストライドを持つ畳み込みである。fが積分である限り、アップサンプリングの自然な方法は、したがって、出力ストライドがfの後方畳み込み(デコンボリューションと呼ばれることもある)である。このような操作は、単に畳み込みの前方パスと後方パスを逆にするだけなので、実装するのは些細なことである。

このように、アップサンプリングは、ピクセル単位の損失からのバックプロパゲーションによるエンドツーエンドの学習のために、ネットワーク内で実行される。

このような層のデコンボリューション・フィルタは(例えばバイリニア・アップサンプリングに)固定する必要はなく、学習できることに注意。デコンボリューション層と活性化関数のスタックは、非線形アップサンプリングを学習することもできる。

我々の実験では、ネットワーク内アップサンプリングが高速で、密な予測を学習するのに有効であることがわかった。我々の最良のセグメンテーションアーキテクチャは、セクション4.2で洗練された予測のためにアップサンプリングを学習するために、これらのレイヤーを使用する。

3.4. パッチワイズ学習は損失サンプリング

確率的最適化では、勾配計算は学習分布によって駆動される。パッチワイズ学習と完全畳み込み学習の両方が、任意の分布を生成するために行うことができるが、それらの相対的な計算効率は、オーバーラップとミニバッチサイズに依存する。画像全体完全畳み込み学習はパッチワイズ学習と同じで、各バッチは画像(または画像の集まり)の損失以下のユニットのすべての受容野から構成される。これはパッチの一様サンプリングよりも効率的であるが、可能なバッチの数を減らすことができる。ただし、画像内のパッチのランダムな選択は、単純に復元することができる。損失を空間項のランダムにサンプリングされた部分集合に制限する(あるいは、同等に出力と損失の間にDropConnectマスク[36]を適用する)ことで、勾配計算からパッチを除外する。

保持されたパッチがまだ大きな重複を持つ場合、完全畳み込み計算は依然として学習を高速化する。複数のバックワードパスにわたって勾配が蓄積される場合、バッチは複数の画像からのパッチを含むことができる[2]。

パッチワイズトレーニングにおけるサンプリングは、クラスの不均衡を修正し[27, 8, 2]、密なパッチの空間相関を緩和することができる[28, 16]。完全畳み込み学習では、損失を重み付けすることでクラスバランスも実現でき、損失サンプリングは空間相関に対処するために使用することができる。

セクション4.3でサンプリングによる学習を検討したが、密な予測に対してより速い、あるいはより良い収束をもたらすことは発見されなかった。画像全体の学習は効果的かつ効率的である。

4. セグメンテーションアーキテクチャ

ILSVRC分類器をFCNにキャストし、ネットワーク内アップサンプリングとピクセル単位の損失で密な予測のために補強する。微調整によりセグメンテーションのための学習を行う。次に、粗い意味情報と局所的な外観情報を組み合わせて予測を洗練させる、新しいスキップアーキテクチャを構築する。

この調査のために、PASCAL VOC 2011 セグメンテーションチャレンジ[7]で訓練と検証を行う。

最終層ユニットの受容野は固定されたストライドグリッド上にあるため、この方法ではすべての可能なパッチが含まれるわけではないことに注意してください。しかし、画像を左右にランダムにストライドまでずらすことで、すべての可能なパッチからランダムに選択することができる。

画素毎の多項ロジスティック損失で学習し、平均画素交差オーバーユニオンの標準的なメトリックで検証する。学習は、グランドトゥルースでマスクされた(曖昧または困難な)ピクセルを無視する。

4.1. 分類器から密なFCNへ

まず、セクション3と同様に、証明された分類アーキテクチャを畳み込むことから始める。ILSVRC12で優勝したAlexNet[3]アーキテクチャ[19]と、ILSVRC14で例外的に良い結果を出したVGGネット[31]とGoogLeNet[4] [32]を検討する。このタスクでは19層のネットと同等であることがわかったVGG 16層のネット[5]を選ぶ。GoogLeNetでは、最終的な損失層のみを使用し、最終的な平均プーリング層を破棄することで性能を向上させる。最後の分類器層を破棄することで各ネットを断頭し、すべての完全連結層を畳み込みに変換する。各粗い出力位置におけるPASCALクラス(背景を含む)のスコアを予測するために、チャンネル次元21の1×1畳み込みを追加し、続いてセクション3.3で説明したように、粗い出力をピクセル密度の高い出力にバイリニアにアップサンプリングするデコンボリューション層を追加する。表1は、各ネットの基本的な特徴とともに、予備的な検証結果を比較したものである。固定学習率(少なくとも175エポック)で収束した後に達成された最良の結果を報告する。

分類からセグメンテーションへの微調整により、各ネットの妥当な予測値が得られた。最悪のモデルでも、最先端の性能の約75%を達成した。セグメンテーション対応VGGネット(FCN-VGG16)は、テスト時の平均IUが52.6であるのに対し、valでは56.0と、すでに最先端であるように見える[16]。余分なデータで学習すると、val[7]のサブセットで59.4平均IUまで性能が上がる。トレーニングの詳細はセクション4.3に示す。

同様の分類精度にもかかわらず、我々のGoogLeNetの実装はこのセグメンテーション結果と一致しなかった。

4.2. 何とどこの組み合わせ

特徴階層の層を組み合わせ、出力の空間精度を向上させる、セグメンテーションのための新しい完全畳み込みネット(FCN)を定義する。図3を参照。

完全畳み込み分類器は、4.1 に示すようにセグメンテーションに微調整することができ、標準的なメトリックでは高いスコアさえ得られるが、その出力は不満足に粗い(図 4 参照)。最終予測層での32ピクセルのストライドは、アップサンプリング出力の詳細のスケールを制限する。

我々は、最終的な予測層と、より細かいストライドを持つ下位層を組み合わせたリンクを追加することで、これに対処する。

表1. 3つの分類コンブネットをセグメンテーションに適応・拡張する。PASCAL VOC 2011の検証セットと推論時間(NVIDIA Tesla K40cの500×500入力に対して20回の試行の平均)で、平均交差和による性能を比較する。パラメータ層の数、出力ユニットの受容野の大きさ、ネット内の最も粗いストライドなど、密な予測に関して適応されたネットのアーキテクチャを詳述する。(これらの数値は、一定の学習率で得られた最高の性能を与えるものであり、最高の性能はありえない)。

	FCN-AlexNet	FCN-VGG16	FCN-GoogLeNet[4]
mean IU	39.8	**56.0**	42.5
forward time	50 ms	210 ms	59 ms
conv. layers	8	16	22
parameters	57M	134M	6M
rf size	355	404	907
max stride	32	32	32

これにより、線トポロジーはDAGに変わり、エッジは下位層から上位層へとスキップ先行する(図3)。ピクセル数が少ないので、より細かいスケールの予測はより少ないレイヤーを必要とするはずなので、より浅いネット出力から作るのは理にかなっている。微細な層と粗い層を組み合わせることで、モデルは大域的な構造を尊重した局所的な予測を行うことができる。Florackら[10]のマルチスケール局所ジェットになぞらえて、我々の非線形局所特徴階層をディープジェットと呼ぶ。

まず、16ピクセルのストライド層から予測することで、出力ストライドを半分に分割する。pool4の上に1×1の畳み込み層を追加し、追加のクラス予測を生成する。この出力と、ストライド32のconv7(畳み込みfc7)の上に計算された予測値を、2×アップサンプリング層を追加し、両方の予測値を合計[6]することで融合する。(図3参照)。2×アップサンプリングをバイリニア補間に初期化するが、セクション3.3で説明したようにパラメータを学習させる。最後に、ストライド16の予測値を画像にアップサンプリングする。これをネットFCN-16sと呼ぶ。FCN-16sはエンド・ツー・エンドで学習され、最後の粗いネットのパラメータで初期化される。pool4に作用する新しいパラメータはゼロ初期化され、ネットは修正されていない予測から始まる。学習率は100分の1に減少している。

このスキップネットを学習することで、検証セットでの性能が3.0平均IU向上し、62.4となった。図4は、出力の微細構造の改善を示している。この融合を、pool4層のみからの学習(性能が低下)、および余分なリンクを追加せずに単純に学習率を低下させた場合(出力の品質を向上させることなく、性能の向上は取るに足らない結果となった)と比較した。

pool3からの予測値をpool4とconv7から融合した予測値の2×アップサンプリングで融合し、ネットFCN-8を構築することで、この方法を続ける。

[3]Using the publicly available `CaffeNet` reference model.

[4]Since there is no publicly available version of GoogLeNet, we use our own reimplementation. Our version is trained with less extensive data augmentation, and gets 68.5% top-1 and 88.4% top-5 ILSVRC accuracy.

[5]Using the publicly available version from the Caffe model zoo.

[6]Max fusion made learning difficult due to gradient switching.

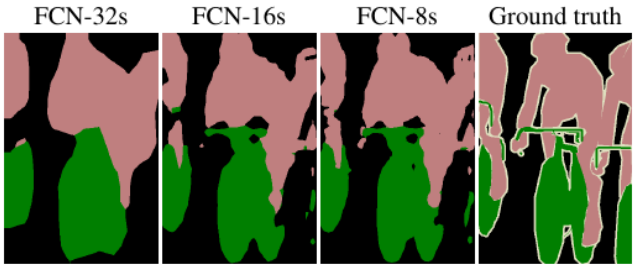

図4. 異なるストライドを持つ層からの情報を融合することで、完全畳み込みネットを洗練させ、セグメンテーションの詳細を改善する。最初の3つの画像は、32、16、8ピクセルのストライドネットからの出力です(図3参照)。

表2. PASCAL VOC2011検証[7]のサブセットにおけるスキップFCNの比較.学習はエンドツーエンドであるが、FCN32s-fixedでは最後の層のみが微調整される。FCN32はFCN-VGG16であり、ストライドを強調するために名前を変更した。

	pixel acc.	mean acc.	mean IU	f.w. IU
FCN-32s-fixed	83.0	59.7	45.4	72.0
FCN-32s	89.1	73.3	59.4	81.4
FCN-16s	90.0	75.7	62.4	83.0
FCN-8s	**90.3**	**75.9**	**62.7**	**83.2**

平均IUは62.7とわずかな改善が見られ、出力の滑らかさとディテールがわずかに改善されていることがわかる。この時点で、我々の融合の改善は、大規模な正しさを強調するIUメトリックに関しても、また、例えば図4に見られるような改善という点でも、収穫逓増を満たしているので、さらに低いレイヤーの融合を続けることはしない。

他の手段による洗練は、プーリング層のストライドを小さくすることが、より細かい予測を得るための最も簡単な方法である。しかし、VGG16ベースのネットでは、そうすることは問題である。pool5層をストライド1に設定すると、受容野サイズを維持するために、畳み込みfc6のカーネルサイズが14×14になる必要がある。

計算コストに加え、このような大きなフィルタの学習が困難であった。pool5以上のレイヤーをより小さなフィルターで再アーキテクトすることを試みたが、同等の性能を達成することには成功しなかった。1つの可能な説明は、上位レイヤーのImageNetで学習した重みからの初期化が重要であるということである。

より細かい予測を得るもう一つの方法は、セクション3.2で説明したシフトアンドスティッチトリックを使用することである。限られた実験では、この方法による改善コスト比はレイヤーフュージョンよりも悪いことがわかった。

4.3. 実験の枠組み

最適化 SGDにより運動量で学習する。FCN-AlexNet、FCN-VGG16、FCN-GoogLeNetはそれぞれ20画像のミニバッチサイズと10^{-3}、10^{-4}、5^{-5}の固定学習率を線探索で選ぶ。運動量0.9、重み減衰5^{-4}または2^{-4}を用い、バイアスの学習率を2倍にしたが、学習はこれらのパラメータに影響されない(しかし学習率には敏感である)ことがわかった。クラススコアリング畳み込み層をゼロ初期化し、より良い性能も収束の速さも得られないランダムな初期化を見つける。元の分類器ネットで使用されたドロップアウトが含まれる。

微調整ネット全体をバックプロパゲーションすることで、全レイヤーを微調整する。出力分類器を単独で微調整しても、表2と比較して、完全な微調整性能の70%しか得られない。基本分類ネットの学習に要する時間を考慮すると、ゼロからの学習は実行不可能である。(VGGネットは段階的に学習されるが、16層の完全バージョンから初期化されることに注意)。FCN-32sの粗いバージョンでは1つのGPUで3日間、FCN-16sとFCN-8sのバージョンにアップグレードするにはそれぞれ約1日かかる。

パッチサンプリング 3.4節で説明したように、我々の完全な画像学習は、各画像を大規模で重なり合ったパッチの規則的なグリッドに効果的にバッチ化する。

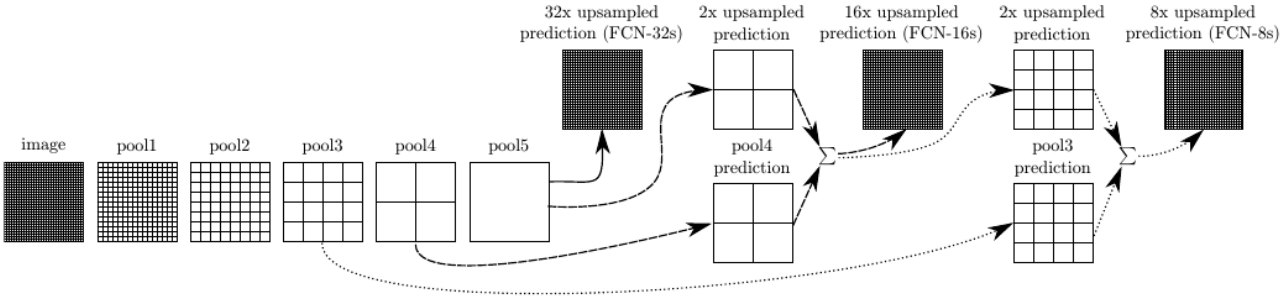

図3. 我々のDAGネットは、粗い高レイヤーの情報と細かい低レイヤーの情報とを組み合わせることを学習する。レイヤーは、相対的な空間粗さを示すグリッドとして表示される。プーリング層と予測層のみを示し、中間畳み込み層(我々の変換した完全連結層を含む)は省略する。実線(FCN-32s): 4.1節で説明したシングルストリームネットは、ストライド32の予測値を1ステップでピクセルにアップサンプリングする。破線(FCN-16s): 最終層と pool4 層の両方からの予測をストライド 16 で組み合わせることで、高レベルの意味情報を保持しながら、より細かいディテールをネットが予測できるようになる。点線(FCN-8s): pool3からの追加予測、ストライド8で、さらなる精度を提供する。

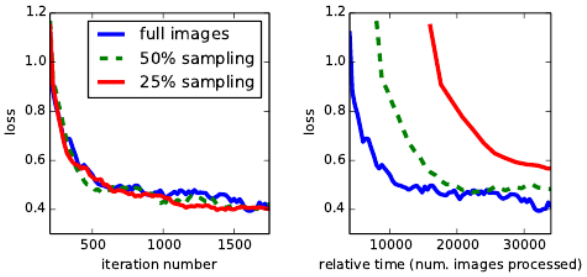

図5. 画像全体に対する学習はパッチのサンプリングと同程度に効果的であるが、より効率的にデータを利用することで、より速い(壁時間)収束をもたらす。左はバッチサイズを固定した場合の収束率に対するサンプリングの効果を示し、右は相対壁時間によって同じプロットをしている。

対照的に、先行研究は、完全なデータセット上でランダムにパッチをサンプリングし[27, 2, 8, 28, 11]、収束を加速する可能性のある高い分散バッチをもたらす可能性がある[22]。我々は、前述の方法で損失を空間的にサンプリングし、ある確率1-pで各最終層セルを無視する独立した選択を行うことによって、このトレードオフを研究する。有効バッチサイズの変更を避けるため、同時にバッチあたりの画像数を1/p倍にする。畳み込みの効率性により、この形式の棄却サンプリングは、pの値が十分に大きい場合(例えば、セクション3.1の数値に従って、少なくともp > 0.2の場合)、パッチワイズ学習よりもまだ高速であることに注意。図5は、この形式のサンプリングが収束に与える影響を示している。サンプリングは全画像学習と比較して収束率に大きな影響を与えないが、バッチごとに考慮する必要がある画像数が多いため、かなり時間がかかることがわかった。したがって、他の実験では、サンプリングされていない全画像学習を選択する。

クラスバランス完全畳み込み学習は、損失を重み付けまたはサンプリングすることで、クラスバランスをとることができる。我々のラベルは軽度のアンバランス(約3/4が背景)であるが、クラスバランシングは不要である。

密な予測ネット内のデコンボリューション層によって、スコアが入力次元にアップサンプリングされる。最終層のデコンボリューション・フィルタはバイリニア補間に固定され、中間層のアップサンプリングはバイリニア・アップサンプリングに初期化され、その後学習される。シフトアンドスティッチ(セクション3.2)、またはフィルタの希薄化に相当するものは使用しない。

Augmentation 各方向に最大32ピクセル(予測値の最も粗いスケール)まで平行移動させることで、ランダムに画像をミラーリングして「ジッタリング」し、学習データの拡張を試みた。この結果、顕著な改善は見られなかった。

より多くのトレーニングデータ PASCAL VOC 2011 セグメンテーションチャレンジのトレーニングセットは、表1で使用し、1112枚の画像をラベル付けした。

Hariharanら[15]は、8498枚のPASCALトレーニング画像の、より大規模なセットのラベルを収集し、これは以前の最先端システムであるSDS[16]のトレーニングに使用された。この学習データにより、FCNVGG16検証スコア[7]は平均IU59.4点に3.4ポイント向上した。

実装全てのモデルは、Caffe [18]を用いて、単一のNVIDIA Tesla K40c上で学習・テストされる。モデルとコードは出版時にオープンソースで公開される。

5. Results

PASCAL VOC、NYUDv2、SIFT Flowを検討し、意味分割とシーン解析でFCNをテストする。これらのタスクは歴史的に物体と領域を区別してきたが、我々は両者を画素予測として一律に扱う。これらのデータセットそれぞれでFCNスキップアーキテクチャ[8]を評価し、NYUDv2のマルチモーダル入力とSIFT Flowの意味的・幾何学的ラベルのマルチタスク予測に拡張する。

指標は、一般的なセマンティックセグメンテーションとシーンパース評価から、ピクセル精度と領域交差オーバーユニオン(IU)のバリエーションである4つの指標を報告する。n_{ij}をクラスjに属すると予測されるクラスiの画素数とし、n_{cl}個の異なるクラスが存在し、$t_i = \sum_j^P n_{ij}$をクラスiの画素の総数とする。計算を行う:

- pixel accuracy: $\sum_i n_{ii} / \sum_i t_i$
- mean accuracy: $(1/n_{\text{cl}}) \sum_i n_{ii}/t_i$
- mean IU: $(1/n_{\text{cl}}) \sum_i n_{ii} / \left(t_i + \sum_j n_{ji} - n_{ii}\right)$
- frequency weighted IU: $\left(\sum_k t_k\right)^{-1} \sum_i t_i n_{ii} / \left(t_i + \sum_j n_{ji} - n_{ii}\right)$

PASCAL VOC 表3は、PASCAL VOC 2011と2012のテストセットにおける我々のFCN-8の性能を示し、以前の最先端技術であるSDS [16]やよく知られたR-CNN [12]と比較したものである。平均IU[9]で20%の相対マージンで最良の結果を達成した。推論時間は114倍(convnetのみ、プロポーザルと絞り込みは無視)または286倍(全体)に短縮される。

表3. 我々の完全畳み込みネットは、PASCAL VOC 2011と2012のテストセットにおいて、最先端技術に対して20%の相対的な改善を与え、推論時間を短縮する。

	mean IU VOC2011 test	mean IU VOC2012 test	inference time
R-CNN [12]	47.9	-	-
SDS [16]	52.6	51.6	~ 50 s
FCN-8s	**62.7**	**62.2**	~ **175 ms**

NYUDv2 [30]は、RGB-Dデータセットを用いて収集された。

[7]There are training images from [15] included in the PASCAL VOC 2011 val set, so we validate on the non-intersecting set of 736 images. An earlier version of this paper mistakenly evaluated on the entire val set.

[8]Our models and code are publicly available at https://github.com/BVLC/caffe/wiki/Model-Zoo#fcn.

[9]This is the only metric provided by the test server.

表4. NYUDv2での結果。RGBDは入力のRGBチャンネルと深度チャンネルの早期融合である。HHAは[14]の深度埋め込みで、水平方向の視差、地上からの高さ、推定された重力方向との局所表面法線の角度として表現される。RGB-HHAは、RGBとHHAの予測値を合計する、共同学習された後期融合モデルである。

	pixel acc.	mean acc.	mean IU	f.w. IU
Gupta *et al.* [14]	60.3	-	28.6	47.0
FCN-32s RGB	60.0	42.2	29.2	43.9
FCN-32s RGBD	61.5	42.4	30.5	45.5
FCN-32s HHA	57.1	35.2	24.2	40.4
FCN-32s RGB-HHA	64.3	44.9	32.8	48.0
FCN-16s RGB-HHA	**65.4**	**46.1**	**34.0**	**49.5**

Microsoft Kinect。Guptaら[13]によって40クラスのセマンティックセグメンテーションタスクに統合されたピクセル単位のラベルを持つ1449枚のRGB-D画像を持つ。795枚のトレーニング画像と654枚のテスト画像の標準的な分割の結果を報告する。(注:すべてのモデル選択はPASCAL 2011 val.で実行) 表4は、いくつかのバリエーションにおける我々のモデルの性能を示している。まず、RGB画像に対して、我々の修正なし粗いモデル(FCN-32s)を学習させる。深度情報を追加するために、4チャンネルのRGB-D入力を取るようにアップグレードされたモデルで学習する(早期融合)。これは、おそらく意味のある勾配をモデル全体に伝播させることが難しいため、ほとんど効果がない。Guptaら[14]の成功に続き、我々は深さの3次元HHAエンコーディングを試し、この情報だけでネットを訓練し、さらにRGBとHHAの「後期融合」を試し、両ネットからの予測が最終層で合計され、結果として得られる2ストリームネットがエンドツーエンドで学習される。最後に、この後期融合ネットを16ストライドバージョンにアップグレードする。

SIFT Flowは、33の意味カテゴリ(「橋」、「山」、「太陽」)と3つの幾何学的カテゴリ(「水平」、「垂直」、「空」)のピクセルラベルを持つ2,688枚の画像のデータセットである。FCNは、両方のタイプのラベルを同時に予測する共同表現を自然に学習することができる。意味的・幾何学的予測層と損失を持つFCN-16sの2頭バージョンを学習する。学習されたモデルは、独立に学習された2つのモデルと同様に両方のタスクで良好な性能を発揮するが、学習と推論は、それぞれ独立したモデルと本質的に同じ速さである。表5の結果は、2,488枚のトレーニング画像と200枚のテスト画像に分割した標準的な[10]で計算したもので、両タスクで最先端の性能を示している。

[10]Three of the SIFT Flow categories are not present in the test set. We made predictions across all 33 categories, but only included categories actually present in the test set in our evaluation. (An earlier version of this paper reported a lower mean IU, which included all categories either present or predicted in the evaluation.)

表5. クラス分割(中央)と幾何学的分割(右)を用いたSIFT Flow[10]の結果。Tighe [33]はノンパラメトリックな転送方法である。ティグ1は模範的なSVMであり、2はSVM+MRFである。Farabetはクラスバランスのとれたサンプル(1)または固有振動数サンプル(2)で学習されたマルチスケールコンブネットである。Pinheiroはマルチスケールリカレントコンブネットであり、R CNN_3 (^{3}) と表記される。ジオメトリの指標はピクセル精度である。

	pixel acc.	mean acc.	mean IU	f.w. IU	geom. acc.
Liu *et al.* [23]	76.7	-	-	-	-
Tighe *et al.* [33]	-	-	-	-	90.8
Tighe *et al.* [34] 1	75.6	41.1	-	-	-
Tighe *et al.* [34] 2	78.6	39.2	-	-	-
Farabet *et al.* [8] 1	72.3	50.8	-	-	-
Farabet *et al.* [8] 2	78.5	29.6	-	-	-
Pinheiro *et al.* [28]	77.7	29.8	-	-	-
FCN-16s	**85.2**	**51.7**	39.5	76.1	**94.3**

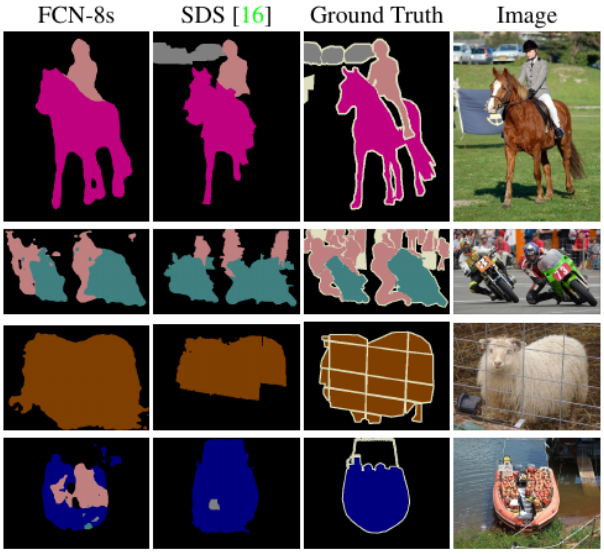

図6. 完全畳み込みセグメンテーションネットは、PASCALにおいて最先端の性能を発揮する。左の列は、最も性能の高いネットであるFCN-8sの出力を示している。2つ目は、Hariharanら[16]による以前の最先端システムによって生成されたセグメンテーションを示す。回収された微細構造(1行目)、密接に相互作用する物体を分離する能力(2行目)、オクルーダーに対する頑健性(3行目)に注目。4行目は失敗例で、ネットはボートの中のライフジャケットを人として見ている。

6. Conclusion

完全畳み込みネットワークはモデルの豊富なクラスであり、その中でも現代の分類畳み込みネットワークは特殊なケースである。このことを認識し、これらの分類ネットをセグメンテーションに拡張し、多解像度レイヤーの組み合わせでアーキテクチャを改善することで、最先端技術を劇的に改善すると同時に、学習と推論を簡素化し、高速化する。

謝辞 DARPAのMSEEおよびSMISCプログラム、

NSFのIIS1427425、IIS-1212798、IIS-1116411、NSFのGRFPであるトヨタ、バークレービジョン・ラーニングセンターから一部支援を受けた。GPUを提供してくれたNVIDIAに感謝する。Bharath HariharanとSaurabh Guptaの助言とデータセットツールに感謝する。CaffeでGoogLeNetを再現してくれたSergio Guadarramaに感謝する。Jitendra Malikの有益なコメントに感謝する。SIFTフローの平均IU計算の問題点と、周波数加重平均IU計算式の誤りを指摘してくれたWei Liuに感謝する。

A. IUの上界

本論文では、粗い意味予測でも平均IUセグメンテーションメトリックで良好な性能を達成した。この指標とそれに対するこのアプローチの限界をよりよく理解するために、様々なスケールでの予測による性能のおおよその上界を計算する。これは、グランドトゥルース画像をダウンサンプリングし、特定のダウンサンプリングファクターで得られる最良の結果をシミュレートするために、再度アップサンプリングすることで行う。次の表は、PASCAL 2011 val のサブセットにおける、様々なダウンサンプリング係数の平均 IU を示している。

factor	mean IU
128	50.9
64	73.3
32	86.1
16	92.8
8	96.4
4	98.5

ピクセルパーフェクト予測は、平均IUが最先端技術をはるかに上回るためには必要でないことは明らかであり、逆に平均IUは微細な精度の良い尺度ではない。

B. More Results

さらに、意味分割のためのFCNを評価する。PASCAL-Context [26]は、PASCAL VOC 2010のシーン全体のアノテーションを提供する。400以上の異なるクラスが存在するが、我々は[26]によって定義された59のクラスタスクに従う。学習セットと評価セットでそれぞれ学習と評価を行う。表6では、このタスクにおける従来の最先端であるConvolutional Feature Masking [3]のjoint object + stuff variationと比較している。FCN-8sは35.1点の平均IUで11%の相対的改善を示した。

Changelog

本論文のarXivバージョンは、修正と追加の関連資料とともに最新に保たれている。以下は、簡単な変更履歴である。

表6. PASCAL-Contextでの結果。CFMは、VGGネットを用いた畳み込み特徴マスキングとセグメント追跡による[3]の最良の結果である。O_2Pは[26]の正誤表で報告されている2次プーリング法[1]である。59クラスのタスクは59の最も頻度の高いクラスを含み、33クラスのタスクは[26]によって識別されたより簡単なサブセットで構成される。

59 class	pixel acc.	mean acc.	mean IU	f.w. IU
O_2P	-	-	18.1	-
CFM	-	-	31.5	-
FCN-32s	63.8	42.7	31.8	48.3
FCN-16s	65.7	46.2	34.8	50.7
FCN-8s	**65.9**	**46.5**	**35.1**	**51.0**
33 class				
O_2P	-	-	29.2	-
CFM	-	-	46.1	-
FCN-32s	69.8	65.1	50.4	54.9
FCN-16s	**71.8**	**68.0**	53.4	57.5
FCN-8s	**71.8**	67.6	**53.5**	**57.7**

v2 平均IUの上界を与える付録Aと、PASCAL-Contextの結果を持つ付録Bを追加する。正しいPASCAL検証番号(以前は、いくつかのval画像がtrainに含まれていた)、SIFT Flow平均IU(不適切に厳密なメトリックを使用)、および周波数加重平均IU式のエラー。モデルへのリンクを追加し、実装の改善を反映してタイミング番号を更新する(一般に公開されている)。

References

[1] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012. 9

[2] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *NIPS*, pages 2852–2860, 2012. 1, 2, 4, 7

[3] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. *arXiv preprint arXiv:1412.1283*, 2014. 9

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014. 1, 2

[5] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 633–640. IEEE, 2013. 2

[6] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014. 2

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes

Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html. 4

[8] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2013. 1, 2, 4, 7, 8

[9] P. Fischer, A. Dosovitskiy, and T. Brox. Descriptor matching with convolutional neural networks: a comparison to SIFT. *CoRR*, abs/1405.5769, 2014. 1

[10] L. Florack, B. T. H. Romeny, M. Viergever, and J. Koenderink. The gaussian scale-space paradigm and the multiscale local jet. *International Journal of Computer Vision*, 18(1):61–75, 1996. 5

[11] Y. Ganin and V. Lempitsky. N^4-fields: Neural network nearest neighbor fields for image transforms. In *ACCV*, 2014. 1, 2, 7

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014. 1, 2, 7

[13] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from RGB-D images. In *CVPR*, 2013. 8

[14] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *ECCV*. Springer, 2014. 1, 2, 8

[15] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 2011. 7

[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision (ECCV)*, 2014. 1, 2, 4, 5, 7, 8

[17] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 1, 2

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 7

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2, 3, 5

[20] Q. V. Le, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng. Building high-level features using large scale unsupervised learning. In *ICML*, 2012. 3

[21] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to hand-written zip code recognition. In *Neural Computation*, 1989. 2, 3

[22] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 1998. 7

[23] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):978–994, 2011. 8

[24] J. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *NIPS*, 2014. 1

[25] O. Matan, C. J. Burges, Y. LeCun, and J. S. Denker. Multi-digit recognition using a space displacement neural network. In *NIPS*, pages 488–495. Citeseer, 1991. 2

[26] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 891–898. IEEE, 2014. 9

[27] F. Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. E. Barbano. Toward automatic phenotyping of developing embryos from videos. *Image Processing, IEEE Transactions on*, 14(9):1360–1371, 2005. 1, 2, 4, 7

[28] P. H. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, 2014. 1, 2, 4, 7, 8

[29] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014. 1, 2, 3, 4

[30] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 7

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1, 2, 3, 5

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. 1, 2, 3, 5

[33] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*, pages 352–365. Springer, 2010. 8

[34] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013. 8

[35] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *CoRR*, abs/1406.2984, 2014. 2

[36] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1058–1066, 2013. 4

[37] R. Wolf and J. C. Platt. Postal address block location using a convolutional locator network. *Advances in Neural Information Processing Systems*, pages 745–745, 1994. 2

[38] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014. 2

[39] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *Computer Vision–ECCV 2014*, pages 834–849. Springer, 2014. 1